

A Hybrid Co-evolutionary Particle Swarm Optimization Algorithm for Solving Constrained Engineering Design Problems

Yongquan Zhou, Shengyu Pei

College of Mathematics and Computer Science Guangxi University for Nationalities, Nanning 530006, China
Email: yongquanzhou@126.com

Abstract—This paper presents an effective hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems, which is based on simulated annealing (SA), employing the notion of co-evolution to adapt penalty factors. By employing the SA-based selection for the best position of particles and swarms when updating the velocity in co-evolutionary particle swarm optimization algorithm. Simulation results based on well-known constrained engineering design problems demonstrate the effectiveness, efficiency and robustness on initial populations of the proposed, and can reach a high precision.

Index Terms—Particle swarm optimization; Simulated annealing; Constrained optimization; Co-evolution; Penalty function

I. INTRODUCTION

Particle swarm optimization (PSO) algorithm [1] is an optimization method widely used to solve continuous nonlinear functions. It is a stochastic optimization technique that was originally developed to simulate the movement of a flock of birds or a group of fish searching for food. Many engineering design problems can be formulated as constrained optimization problems. So far, penalty function methods have been the most popular methods for constrained optimization due to their simplicity and easy implementation. However, it is often not easy to set suitable penalty factors or to design adaptive mechanism. Distinguish from penalty function method; co-evolutionary particle swarm optimization approach (CPSO) [2] is an effective method for solving constrained optimization problems.

Particle swarm optimization (PSO) is a novel population-based searching technique proposed in 1995 as an alternative to genetic algorithm (GA) [3]. Compared with GA, PSO has some attractive characteristics. Firstly, PSO has memory, that is, the knowledge of good solutions is retained by all particles, whereas in GA, previous knowledge of the problem is destroyed once the population changes. Secondly, PSO has constructive cooperation between particles, that is, particles in the

swarm shares their information. On the other hand, similar to GA, it is also shown that PSO is often easy to be premature convergence so that exploration (searching for promising solutions within the entire region) and exploitation (searching for improved solutions in sub-regions) should be enhanced and well balanced to achieve better performance. Thus, we will consider the development of more effective CPSO-based approach. In this paper, we will propose an effective hybrid strategy named CPSOSA by incorporating the jumping mechanism of simulated annealing (SA) [4], [5] into CPSO to achieve results with high quality and reliance.

The remaining contents are organized as follows. In Section 2, the problem statement of constrained engineering design problems is provided. In Section 3, the hybrid CPSOSA strategy is proposed and explained in detail. Simulation results can be found in Section 4. Finally, in Section 5, we end the paper with some conclusions and future work.

II. PROBLEM STATEMENT OF CONSTRAINED ENGINEERING DESIGN

Generally, a constrained optimization problem can be described as follows:

Find x to Minimize $f(x)$

Subject to: $g_i(x) \leq 0; i = 1, 2, \dots, n$. (1)

$$h_j(x) = 0, j = 1, 2, \dots, p$$

where $x = [x_1, x_2, \dots, x_d]^T$ denotes the decision solution vector, n is the number of inequality constraints and p is the number of equality constraints. In a common practice, equality constraint $h_j(x) = 0$ can be replaced by a set of inequality constrained $h_j(x) \geq \delta$ and $h_j(x) \leq -\delta$ (δ is a small tolerant amount). Thus, all constraints can be transformed to $N = n + 2p$ inequality constraints.

Many engineering design problems can be formulated as constrained optimization problems. The presence of constraints may significantly affect the optimization

*Corresponding author. Tel. +86 771 3260264

E-mail: yongquanzhou@126.com (Yongquan Zhou)

performances of any optimization algorithms for unconstrained problems. For most constraint-handling techniques, both infeasible and feasible solutions could be generated at the search stage, and constraints are dealt with when evaluating solutions. The violation of constraints for each solution is considered separately, and the relationship between infeasible solutions and feasible regions is exploited to guide search. The difficulty lies in that there is no certain metric criterion to measure this relationship. Distinguish from penalty function method; a co-evolutionary particle swarm optimization approach (CPSO) is proposed for constrained optimization problems by applying PSO and employing the notion of co-evolution (Wang, 2006).

III. CPSOSA HYBRID STRATEGY

A. The standard PSO

The PSO is first introduced by Kennedy and Eberhard, it is a stochastic optimization technique that can be likened to the behavior of a flock of birds or the sociological behavior of group problems, including neural network training and function minimization. Several attempts have made to improve the performance of the original PSO, some of which are discussed in this section.

Let s denote the swarm size, each individual $1 \leq i \leq s$ has the following attributes. A current position in the search spaces $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,s}]^T$, a current velocity $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,s}]^T$; each particle has its own best position (pbest) $Y_i = [y_{i,1}, y_{i,2}, \dots, y_{i,s}]^T$ corresponding to the personal best objective value obtained so far at time t . The global best particle (gbest) is denoted by \hat{y}_j . During each iteration, each particle in the swarm is updated using (2) and (3). Assuming that the function f is to be minimized, that the swarm consists of n particles, and that $r_1 \sim U(0,1)$, $r_2 \sim U(0,1)$ are elements from two uniform random sequences in the range of $(0,1)$, then

$$\begin{aligned} v_{i,j}(t+1) &= wv_{i,j}(t) + c_1r_1[y_{i,j}(t) - x_{i,j}(t)] \\ &\quad + c_2r_2[\hat{y}_j(t) - x_{i,j}(t)] \end{aligned} \quad (2)$$

For all $j \in \{1, 2, \dots, n\}$, where $v_{i,j}$ is the velocity of the j th dimension of the i th particles, c_1 and c_2 are constants called acceleration coefficients, w is called the inertia factor. The new position of a particle is calculated using

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), j = 1, 2, \dots, n. \quad (3)$$

The personal best position of each particle is updated using equation

$$Y_i(t+1) = \begin{cases} Y_i(t), & \text{if } f(X_i(t+1)) \geq f(Y_i(t)) \\ X_i(t+1), & \text{if } f(X_i(t+1)) < f(Y_i(t)) \end{cases} \quad (4)$$

and the global best position found by any particle during all previous steps, \hat{y} is defined as

$$\hat{y}(t+1) = \arg \min_{Y_i} f(Y_i(t+1)), 1 \leq i \leq s. \quad (5)$$

B. CPSO strategy

The principle of co-evolution model in CPSO is shown in Figure 1. In CPSO, two kinds of swarms are used. In particular, one kind of a single swarm (denoted by $Swarm_2$) with size M_2 is used adapt suitable penalty factors, another kind of multiple swarms (denoted by $Swarm_{1,1}, Swarm_{1,2}, \dots, Swarm_{1,M_2}$) each with size M_1 are used in parallel to search good decision solutions. Each particle B_j in $Swarm_2$ represents a set of penalty factors for particles in $Swarm_{1,j}$, where each particle represents a decision solution.

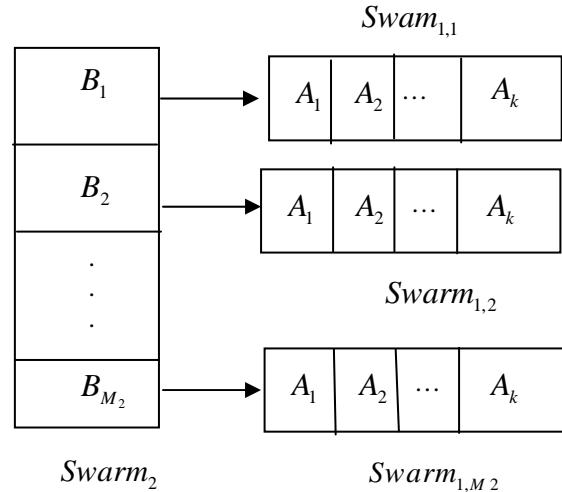


Figure 1. Graphical illustration for notion of co-evolution.

In every generation of co-evolution process, every $Swarm_{1,j}$ will evolve by using PSO for a certain number of generations G_1 with particle B_j in $Swarm_2$ as penalty factors for solution evaluation to get a new $Swarm_{1,j}$. Then the fitness of each particle B_j in $Swarm_2$ will be determined. After all particles in $Swarm_2$ are evaluated, $Swarm_2$ will also evolve by using PSO with one generation to get a new $Swarm_2$ with adjusted penalty factors. The above co-evolution process will be repeated until a pre-defined stopping criterion is satisfied (e.g., a maximum number of co-evolution generations G_2 are reached).

In particular, the i th particle in $Swarm_{1,j}$ in CPSO is evaluated by using the following formula:

$$F_i(x) = f_i(x) + sum_viol \times w_1 + num_viol \times w_2, \quad (6)$$

where $f_i(x)$ is the objective value of the i th particle, sum_viol denotes the sum of all the amounts by which the constraints are violated, num_viol denotes the number of constraints violation, w_1 and w_2 are penalty factors corresponding to the particle B_j in $Swarm_2$.

The value of sum_viol is calculated as follows:

$$sum_viol = \sum_{i=1}^N g_i(x), \quad \forall g_i(x) > 0 \quad (7)$$

where N is the number of inequality constraints (here it is assumed that all equality constraints have been transformed to inequality constraints).

Each particle in $Swarm_2$ represents a set of factors (w_1 and w_2). After $Swarm_{1,j}$ evolves for a certain number of generations (G_1), the j th particle B_j in $Swarm_2$ is evaluated as follows.

(1) If there is at least one feasible solution in $Swarm_{1,j}$, then particle B_j is evaluated using the following formula and is called a valid particle:

$$P(B_j) = \frac{\sum f_{feasible}}{num_feasible} - num_feasible, \quad (8)$$

Where $\sum f_{feasible}$ denotes the sum of objective function values of feasible solutions on $Swarm_{1,j}$, and $num_feasible$ is the number of feasible solutions in $Swarm_{1,j}$.

(2) If there is no feasible solution in $Swarm_{1,j}$ (it can be considered that the penalty is too low), then particle B_j in $Swarm_2$ is evaluated as follows and is called an invalid particle.

$$P(B_j) = \max(P_{valid}) + \frac{\sum sum_viol}{\sum num_viol} - \sum num_viol, \quad (9)$$

where $\max(P_{valid})$ denotes the maximum fitness value of all valid particles in $Swarm_2$, $\sum sum_viol$

denotes the sum of constrains violation for all particles in $Swarm_{1,j}$, and $\sum num_viol$ counts the total number of constraints violation for all particles in $Swarm_{1,j}$.

The particle in $Swarm_2$ encodes a set of penalty factors (w_1 and w_2), while the particle in $Swarm_{1,j}$ encodes a set of decision variables. Both kinds of particles will apply Eqs. (2) and (3) to adjust their positions so as to obtain good decision solution and suitable penalty factors.

C. Hybrid CPSOSA strategy

The main deficiency of PSO is easy to be premature convergence. One reason is that, in PSO \hat{y}_g is used as

\hat{y}_i in Eq. (5). Consequently, all particles have the tendency to fly to the current best solution that may be a local optimum or a solution near local optimum, so that all particles will concentrate to a small region and the global exploration ability will be weakened. That is to say, if \hat{y}_g is not the global optimum, the algorithm evolved with Eqs. (2) and (3) may miss the region containing the global optimum or may trap in a local optimum.

As we known, simulated annealing is a stochastic searching algorithm with jumping property motivated by the similarity between the solids' annealing procedure and optimization problems. The most significant character of SA is the probabilistic jumping property, i.e., a worse solution has a probability to be accepted as the new solution. Moreover, by adjusting the temperature, such a jumping probability can be controlled. In particular, the probability is rather high when temperature is high and decreases as the temperature decreases; and when the temperature tends to zero the probability approaches to zero so that only better solutions can be accepted. It has been theoretically proved that under certain conditions SA is globally convergent in probability 1. In this section, we try to incorporate the mechanism of SA into CPSO to propose a hybrid optimization strategy, named CPSOSA.

As mentioned before, in CPSO \hat{y}_g is one element of the set of all y_i , which can be regarded as a set of "local optima". Thus, we attempt to modify the selection of \hat{y}_g to overcome premature convergence. In particular, we employ the cross factor of SA. That is, we do use one of y_i (denoted by \hat{y}_g) as \hat{y}_i . Borrowing the mechanism of SA, all other y_i can be regarded as special solutions

Worse than \hat{y}_i , and we define $e^{(F\hat{y}_g - F\hat{y}_i)/t}$ as the fitness value of each y_i to replace \hat{y}_i at a certain temperature t .

The whole procedure of CPSOSA is described as follows:

Step 1: (Initialization):

Initialize $Swarm_1$ and $Swarm_2$, and evaluate particles in $Swarm_1$. Duplicate $Swarm_1$; M_2 copies as $Swarm_{1,1}, Swarm_{1,2}, \dots, Swarm_{1,M_2}$. Let $l = 0$, $t = 0$, $b = 0$, $temperature = 1e-8$, $decayScale = 0.7$.

Step 2: Repeat until a stopping criterion is satisfied ($l = G_2$):

Step 2.1: Repeat until a stopping criterion is satisfied ($t = G_1$);

Step 2.1.1: Evolve $Swarm_{1,j}$ ($j = 1, 2, \dots, M_2$) using PSOSA with penalty factors B_j ($j = 1, \dots, M_2$).

Step 2.1.2: $t = t + 1$;

Step 2.2: Calculate fitness of all particles B_i in $Swarm_2$ ($i = 1, 2, \dots, M_2$).

Step 2.3: Evolve $Swarm_2$ using PSO with one generation to get a new $Swarm_2$ with adjusted penalty factors.

Step 2.4: Let $l = l + 1$ and $t = 0$;

Step 3: Output the best $gbest$ of all $Swarm_{1,j}$.

Lastly, the proposed CPSOSA is a general optimization algorithm that can be applied to any constrained optimization problems. In the next section, we will apply such an approach for constrained engineering design problems.

IV. SIMULATION RESULTS AND ANALYSIS

In this section, we will carry out numerical simulation based on some well-known constrained engineering design problems to investigate the performances of the proposed CPSOSA.

For each testing problem, the parameters of the CPSOSA are set as follows: $M_1 = 50$, $G_1 = 25$, $G_2 = 10$, $M_2 = 20$, $c_1 = c_2 = 2.0$, w in PSO linearly decrease from 0.9 to 0.4. The maximum and minimum positions for particles in $Swarm_{1,j}$ ($X_{1,max}$ and $X_{1,min}$) depend on the variable region given by the problems. The maximum and minimum positions of particles in $Swarm_2$ are set as $w_{1,max} = w_{2,max} = 1000$ and $w_{1,min} = w_{2,min} = 0$ for the first two problems and as $w_{1,max} = w_{2,max} = 10000$ and $w_{1,min} = w_{2,min} = 5000$ for the third problem. Moreover, the maximum and minimum velocities for

particles in both the kinds of swarms are set as $V_{i,max} = 0.2 \times (X_{i,max} - X_{i,min})$ and $V_{i,min} = -V_{i,max}$ ($i = 1, 2$).

A. *Simulation results for a welded beam design problem (Example 1)*

The welded beam design problem is taken from Rao (1996), in which a welded designed for minimum cost subject to constraints on shear stress (τ), bending stress in the beam (θ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constrains. There are four design variables as shown in Figure 2, i.e. $h(x_1)$, $l(x_2)$, $t(x_3)$, and $b(x_4)$.

The problem can be mathematically formulated as follows:

Minimize

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 - x_2) \quad (10)$$

subject to :

$$g_1(x) = \tau(x) - 13000 \leq 0,$$

$$g_2(x) = \sigma(x) - 30000 \leq 0,$$

$$g_3(x) = x_1 - x_4 \leq 0,$$

$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0,$$

$$g_5(x) = 0.125 - x_1 \leq 0,$$

$$g_6(x) = \delta(x) - 0.25 \leq 0,$$

$$g_7(x) = 6000 - P_c(x) \leq 0,$$

Where

$$\tau(x) = \sqrt{(\tau')_2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$

$$\tau' = \frac{6000}{\sqrt{2}x_1x_2},$$

$$\tau'' = \frac{MR}{J},$$

$$M = 6000(14 + \frac{x_2}{2}),$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2},$$

$$J = 2 \left\{ \sqrt{2}x_1x_2[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2] \right\},$$

$$\sigma(x) = \frac{504000}{x_4x_3^2},$$

$$\delta(x) = \frac{2.1952}{x_3^3x_4},$$

$$P_c(x) = 64746.022(1 - 0.0282346x_3)x_3x_4^3$$

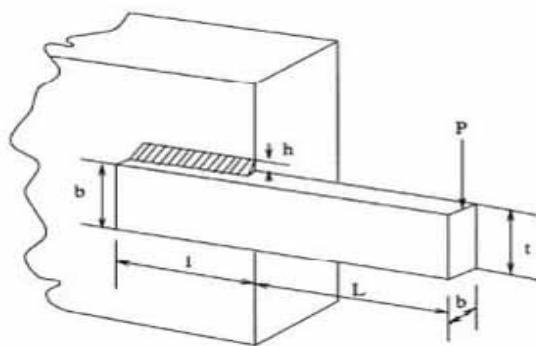


Figure 2. A welded beam design problem (Example 1).

The approaches applied to this problem include geometric programming (Ragsdell and Phillips, 1976), genetic algorithm with binary representation and traditional penalty function (Deb, 1991), a GA-based co-

evolution model (Codillo, 2000), a feasibility-based tournament selection scheme inspired by the multi-objective optimization techniques (Coello and Montes, 2000), an effective co-evolutionary particle swarm optimization for constrained engineering design problems (Qie He, Ling Wang, 2006), and a hybrid particle swarm optimization with a feasibility-based rule for constrained optimization (He Q, Wang L, 2007). In this paper, the CPSOSA is run 30 times independently with the following variable regions: $0.1 \leq x_1 \leq 2$, $0.1 \leq x_2 \leq 10$, $0.1 \leq x_3 \leq 10$, $0.1 \leq x_4 \leq 2$. The best solutions obtained by the abovementioned approaches are listed in Table 1, and their statistical simulation results are shown in Table 2.

Table 1 Comparison of the best solution of Example 1 found by different methods

Design variables	CPSOSA	CPSO (2006)	HPSO (2007)	Ragsdell and Phillips (1976)	Deb (1991)	Coello (2000)	Coello and Montes (2002)
$x_1(h)$	0.205729615134341	0.202369	0.205730	0.245500	0.248900	0.208800	0.205986
$x_2(l)$	3.470489004516055	3.544214	3.470489	6.196000	6.173000	3.420500	3.471328
$x_3(t)$	9.036624454433026	9.048210	9.036624	8.273000	8.178900	8.997500	9.020224
$x_4(b)$	0.205729619072119	0.205723	0.205730	0.245500	0.253300	0.210000	0.206480
$g_1(x)$	-5.550180139834993e-005	-12.839796	1.724852	-5743.826517	-5758.603777	-0.337812	-0.074092
$g_2(x)$	-5.919082323089242e-004	-1.247467	N/A	-4.715097	-255.576901	-353.902604	-0.266227
$g_3(x)$	-3.937778270124426e-009	-0.001498	N/A	0.000000	-0.004400	-0.001200	-0.000495
$g_4(x)$	-3.432983819363303	-3.429347	N/A	-3.020289	-2.982866	-3.411865	-3.430043
$g_5(x)$	-0.080729615134341	-0.079381	N/A	-0.120500	-0.123900	-0.083800	-0.080986
$g_6(x)$	-0.235540323740633	-0.235536	N/A	-0.234208	-0.234160	-0.235649	-0.2335514
$g_7(x)$	-2.411008881608723e-005	-11.681355	N/A	-3604.275002	-4465.270928	-363.232384	-58.666440
$f(x)$	1.724852	1.728024	1.724852	2.385937	2.433116	1.748309	1.728226

Table 2 Statistical results of different methods for Example 1

Method	Best	Mean	Worst	Std Dev
CPSOSA	1.724852	1.724853	1.724861	1.705146210873380e-005
CPSO(2006)	1.728024	1.748831	1.782143	0.01926
Belegundu(1982)	2.385937	N/A	N/A	N/A
Arora(1989)	2.433116	N/A	N/A	N/A
Coello(2000)	1.748309	1.771973	1.785835	0.011220
Coello and Montes(2002)	1.728226	1.792654	1.993408	0.074713

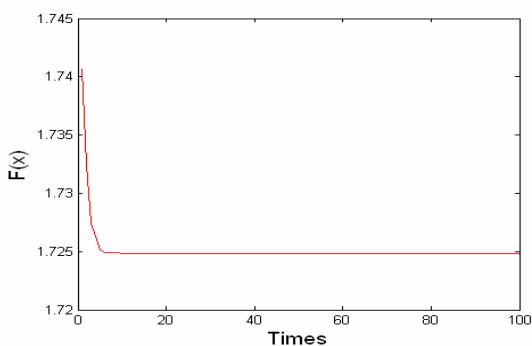


Figure 3. Evolution of $gbest$ of $Swarm_{1,j}$

This problem is from Arora (1989) and Belegundu (1982), which needs to minimize the weight (i.e. $f(x)$) of a tension/compression string (as shown in Figure 4) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter $D(x_2)$, the wire diameter $d(x_1)$ and the number of active coils $P(x_3)$.

The mathematical formulation of this problem can be described as follows:

$$\text{Minimize } f(x) = (x_3 + 2)x_2 x_1^2 \quad (11)$$

subject to :

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$

Table 3 Comparison of the best solution for Example 2 by different methods

Design variables	CPSOSA	CPSO (2006)	HPSO (2007)	Belegundu (1982)	Arora (1989)	Coello (2000)	Coello and Montes (2002)
$x_1(d)$	0.051653711770636	0.051728	0.051706	0.050000	0.053396	0.051480	0.051989
$x_2(D)$	0.355867916029741	0.357644	0.357126	0.315900	0.399180	0.351661	0.363965
$x_3(P)$	11.338963731041684	11.244543	11.265083	14.250000	9.185400	11.632201	10.890522
$g_1(x)$	-6.310567490785957e-008	-0.000845	N/A	-0.000014	0.000019	-0.002080	-0.000013
$g_2(x)$	-1.740363408941903e-008	-1.2600e-05	N/A	-0.003782	-0.000018	-0.000110	-0.000021
$g_3(x)$	-4.052103533284539	-4.051300	N/A	-3.938302	-4.123832	-4.026318	-4.061338
$g_4(x)$	-0.728318914799749	-0.727090	N/A	-0.756067	-0.698283	-4.026318	-0.72698
$f(x)$	0.01266525	0.0126747	0.0126652	0.0128334	0.0127303	0.0127048	0.0126810

From Figure 3, we can find that, compared with the results obtained by CPSOSA, it is demonstrated that CPSOSA is of effectiveness to avoid being trapped in local optima by incorporating the jumping mechanism of SA into pure CPSO.

B. Simulation results for a tension/compression string design problem (Example 2)

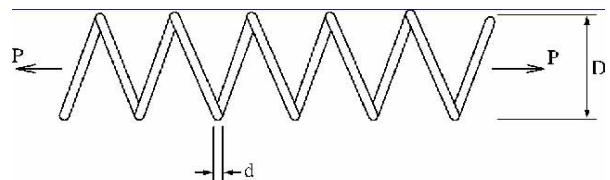


Figure 4. A tension/compression string design problem (Example 2).

The approaches applied to this problem include six different numerical optimization techniques (Belegundu, 1982), a numerical optimization technique called constraint correction at constant cost (Arora, 1989), a GA-based co-evolution model (Coello, 2000) and a feasibility-based tournament selection scheme (Coello and Montes, 2002), an effective co-evolutionary particle swarm optimization for constrained engineering design problems (Qie He, Ling Wang, 2006), and a hybrid particle swarm optimization with a feasibility-based rule for constrained optimization (He Q, Wang L, 2007). In this paper, the CPSOSA is run 30 times independently with the following variable regions: $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$. The best solutions obtained by the above-mentioned approaches are listed in Table 3, and their statistical simulation results are shown in Table 4.

From Table 3, it can be seen that the best feasible solution found by CPSOSA is better than the best solutions found by other techniques

Table 4 Statistical results of different methods for Example 2

Method	Best	Mean	Worst	Std Dev
CPSOSA	0.01266525	0.012668	0.012671	3.685693653588679e-006
CPSO(2006)	0.0126747	0.012730	0.012924	5.198500e-005
Belegundu(1982)	0.0128334	N/A	N/A	N/A
Arora(1989)	0.0127303	N/A	N/A	N/A
Coello(2000)	0.0127048	0.012769	0.012822	3.939000e-005
Coello and Montes(2002)	0.0126810	0.0127420	0.012973	5.9000000e-005

C. Simulation results for a pressure vessel design problem (Example 3)

In this problem, the objective is to minimize the total cost ($f(x)$), including the cost of the material, forming and welding. A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 5. There are four design variables: T_s (x_1 , thickness of the shell), T_h (x_2 , thickness of the head), R (x_3 , inner radius) and L (x_4 , length of the cylindrical section of the vessel, not including the head). Among the four variables, T_s and T_h are integer multiples of 0.0625 in that the available thickness of rolled steel plates, and R and L are continuous variables.

The problem can be formulated as follows (Kannan and Karmer, 1994):

Minimize

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3x_2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (12)$$

subject to:

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1296000 \leq 0,$$

$$g_4(x) = x_4 - 240 \leq 0.$$

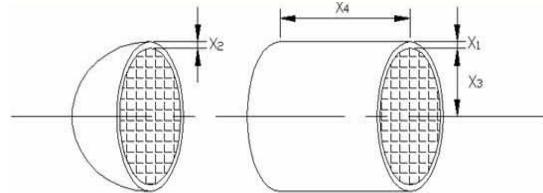


Figure 5. Center and end section of pressure vessel design problem (Example 3).

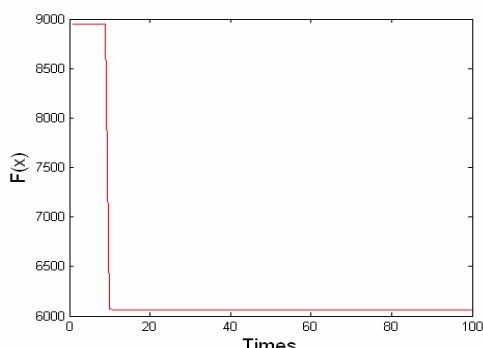
The approaches applied to this problem include genetic adaptive search (Deb, 1997), an augmented Lagrangian multiplier approach (Kannan and Kramer, 1994), a branch and bound technique (Sandgren, 1988), a GA-based co-evolution model (Coello, 2000) and a feasibility-based tournament selection scheme (Coello and Montes, 2002), an effective co-evolutionary particle swarm optimization for constrained engineering design problems (Qie He, Ling Wang, 2006), and a hybrid particle swarm optimization with a feasibility-based rule for constrained optimization (He Q, Wang L, 2007). In this paper, the CPSOSA is run 30 times independently with the following variable regions: $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10 \leq x_3 \leq 200$, $10 \leq x_4 \leq 200$. the best solutions obtained by the above mentioned approaches are listed in Table 5, and their statistical simulation results are shown in Table 6.

Table 5 Comparison of the best solution for Example 3 found by different methods

Design variables	CPSOSA	CPSO (2006)	HPSO (2007)	Standgren (1988)	Kannan and Kramer (1994)	Deb (1997)	Coello (2000)	Coello and Montes (2002)
$x_1(T_s)$	0.812500000000000	0.812500	0.8125	1.125000	1.125000	0.937500	0.812500	0.812500
$x_2(T_s)$	0.437500000000000	0.437500	0.4375	0.625000	58.291000	0.500000	0.437500	0.437500
$x_3(T_s)$	42.098445593492706	42.091266	42.0984	47.700000	43.690000	48.329000	40.323900	42.097398
$x_4(T_s)$	1.766365958720004e+002	176.746500	176.6366	117.70100	0.000016	112.679000	2000.0000	176.65405
$g_1(x)$	-4.559075339471974e-011	-0.000139	N/A	-0.204390	-0.068904	-0.004750	-0.034324	-0.000020
$g_2(x)$	-0.035880829038080	-0.035949	N/A	-0.169942	-0.068904	-0.038941	-0.052847	-0.035891
$g_3(x)$	-1.611420884728432e-006	-116.382700	N/A	54.226012	-21.220104	3652.876838	27.105845	-27.886075
$g_4(x)$	-63.363404127999559	-63.253500	N/A	-122.29900	-196.310000	-127.321000	-40.00000	-63.345953
$f(x)$	6059.7143	6061.0777	6059.7143	8129.1036	7198.0428	6410.3811	6288.7445	6059.9463

Table 6 Statistical results of different methods for Example 3

Method	Best	Mean	Worst	Std Dev
CPSOSA	6059.7143	6059.7143	6059.7146	2.264154757866423e-006
CPSO(2006)	6061.0777	6147.1332	6363.8041	86.4545
Sandgren(1988)	8129.1036	N/A	N/A	N/A
Kannan and Kramer(1994)	7198.0428	N/A	N/A	N/A
Deb(1997)	6410.3811	N/A	N/A	N/A
Coello(2000)	6288.7445	6293.8432	6308.1497	7.4133
Coello and Montes(2002)	6059.9463	6177.2533	6469.3220	130.9297

**Figure 6.** Evolution of *gbest* of $Swarm_{1,j}$

From Figure 6, we also found that, compared with the results obtained by CPSOSA, it is demonstrated that CPSOSA is of effectiveness to avoid being trapped in local optima by incorporating the jumping mechanism of SA into pure CPSO.

Based on the above simulation results and comparisons, it can be concluded that CPSOSA is of superior searching quality and robustness for constrained engineering design problems.

V. CONCLUSIONS

This paper has introduced a novel constraint-handling method-CPSOSA. This is the first report to using CPSOSA for constrained engineering design problems, which incorporate the mechanism of SA into CPSO. Simulation results and comparisons based on some well-known constrained engineering design problems and comparisons with previously reported results demonstrate the effectiveness, efficiency and robustness of CPSOSA. The future work is to investigate better mechanism into CPSOSA to achieve better performance. Our future work is to study the parallel implementation of CPSOSA and the application of CPSOSA for constrained combinatorial optimization problems.

ACKNOWLEDGMENT

This work is supported by Grants 60461001 from NSF of China and the project Supported by Grants 0832082, 0991086 from Guangxi Science Foundation

REFERENCES

- [1] R.C. Eberhart and J. Kennedy. "A new optimizer using particle swarm theory." In Proc. 9th Int. Conf. Neural Information Processing, Nov.2002, pp. 1140-1145.
- [2] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation, 2007, 186(2): 1407-1422.
- [3] D.E. Goldberg. Genetic Algorithms in Search Intelligence, Machine Learning. Addison Wesley, MA, 1989.
- [4] L. Wang. Intelligent Optimization Algorithms with Applications. Tsinghua University and Springer Press, Beijing, 2001.
- [5] LI Ling-Lai; LING Wang; Li-Heng. An effective hybrid PSOSA strategy for optimization and its application to parameter estimation. Applied Mathematics and Computation, 179(1):135-146. 2006.
- [6] Carlos A. Coello Coello, Efrén Mezura Montes. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Advanced Engineering Informatics, 2002.16(3):193-203.
- [7] Deb, K., Optimal design of a welded beam via genetic algorithms. AIAA Journal, 1991, 29 (11): 2013-2015.
- [8] Kannan, B.K., Kramer, S.N., An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. Journal of Mechanical Design, 1994. 116(2): 405-412.
- [9] Montes, E.M., Coello, C.A.C., A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary computation. 2005, 9(1):1-17.
- [10] Rao, S.S. Engineering Optimization. Wiley, New York. 1996.
- [11] Sandgren, E., 1988. Nonlinear integer and discrete programming in mechanical design. In: Proceedings of the ASME Design Technology Conference, Kissimmee, FL, pp. 95-105.

Yongquan Zhou Ph.D. Professor. His received the B.S. degree in mathematics from Xianyang Normal University, Xianyang, China. In 1983. the M.S. degree in computer from Lanzhou University , Lanzhou, China. In 1993, and Ph.D. in computation intelligence from Xidian University, Xian, China. His current research interests including computation intelligence and applies.

Shengyu Pei received the M.S. degree in computer from Guangxi University for Nationalities, Nanning, China. He current research interests including computation intelligence and applies